LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

UCRL-TR-206057

# Matching Shapes Using Local Descriptors

R. White, S. Newsam, C. Kamath

August 16, 2004

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

# Matching Shapes Using Local Descriptors

Ryan White

Department of Electrical Engineering and Computer Science

University of California, Berkeley

ryanw@cs.berkeley.edu

Shawn Newsam and Chandrika Kamath

Center for Applied and Scientific Computing

Lawrence Livermore National Laboratory

newsam1@llnl.gov and kamath2@llnl.gov

**Abstract**

We present a method for comparing shapes of grayscale images in noisy circumstances. By establishing correspondences in a new image with a shape model, we can estimate a transformation between the new region and the model. Using a cost function for deviations from the model, we can rank resulting shape matches. We compare two separate distinct region detectors: Scale Saliency and difference of gaussians. We show that this method is successful in comparing images of fluid mixing under anisotropic geometric distortions and additive gaussian noise. Scale Saliency outperforms the Difference of Gaussians in this context.

## 1 Introduction

Shape matching is a branch of computer vision / image processing that has received considerable attention in the past. The typical approach to this field is to work with a black and white image of a discrete shape. While it is possible to convert grayscale images to black and white images for the purpose of shape analysis, the kind of noise generated in this process is exactly the weak point of shape analysis in this field. Additionally, the work in this field focuses on complete unoccluded images.

Instead of throwing away grayscale information, we seek to use the grayscale information to our advantage in order to analyze shape. Specifically, our approach focuses on the grayscale information for its descriptive power. By using commonly available feature descriptors to describe grayscale regions of an image, we can establish correspondences between two images of a similar object. With correspondences between the images established, we can estimate a geometrical transformation between the two images and penalize changes in shape that don't

fit our model.

Our application domain is images of fluid mixing - images that contain significant information in the grayscale. We test our method in several ways: we perform a search through our database of images to find similar shapes, we artificially alter the images by geometric trasformations and by adding noise.

# 2 Background

## 2.1 Shape Analysis

Previous work in shape analysis focuses on describing black and white images [12]. Contour based methods extract a contour of the image region and compute a transformation between contours. These methods are particularly ill suited towards real world shape comparison because they don't deal well with occlusion or noise. The Hausdorff distance effectively computes the maximum min distance between two contours / regions / point clouds. This has huge dis-advantages in dealing with noise - a single point in the wrong place completely destroys the metric.

A more suitable technique involves computing the area of overlap between the two regions of dark pixels. While it can deal with small occlusions and spurious points well, it suffers from poor robustness to common image transformations such as rotation and scale. While not computationally efficient, one could search over scales, rotations and translations to mitigate these problems. However, this would still not meet our needs in two repects: overcoming large occlusions and utilizing grayscale information.

Other techniques in shape recognition focus on describing the region and using the descriptor for comparison. Zernike moments [5], a method of projecting the image onto a polynomial basis, outperform simpler histogram techniques [13]. While better than area of overlap, Zernike moments don't deal well with image transformations or occlusion. While it seems reasonable to assume that grayscale images could be projected onto this basis, we found little evidence of this in the literature.

Drawing from the computer vision community, shape context seems to be a common shape descriptor [1] [8]. The image is first converted to a cloud of points on the edge of the shape and then each point gets a histogram descriptor that is radially symmetric and logarithmically spaced outward. A continuous warping of an image has only a small effect on the local structure but can have a profound effect on the global structure. Shape context takes advantage of this by recording local structure much more accurately than global structure. Again, this doesn't address the issue of shape in a grayscale image.

Another technique involves comparing shock graphs (a slightly more advanced version of a medial axis transform) [11]. In that work, an edit distance is designed for shock graphs. Again, shock graphs aren't robust to common forms of noise, and don't easily deal with grayscale images.

Finally, in the object recognition literature, Rothganger et al. [9] propose a method using SIFT (Scale Invariant Featuer Transform) features to build 3D models of objects and compare them. Although they attempt

to solve a different problem, their work is similar to ours.

## 2.2 Distinct Region Detectors

We surveyed a number of difference techniques to find distinct regions in images. Our goal in picking a region detector is twofold. First, we need to identify enough regions to build a resonable model of shape (and correspondingly hope that they cover a significant portion of the image). Second, we need a region detector that is consistant in identifying features across common imaging transformations including scale, rotation and translation.

We can divide common region detectors into two categories: those that detect circles and those that detect ellipses. In 3D imagery, ellipses are helpful because they provide an affine description of the region. Of the detectors that we could easily find in the literatue, only [10] detected affine regions. Using this detector on our fluid mixing images produced a small number of regions - in some cases not a single response in our query region. Since our method requires a larger number of regions, and our images are 2D and don't require affine information, we elected to not pursue this method further.

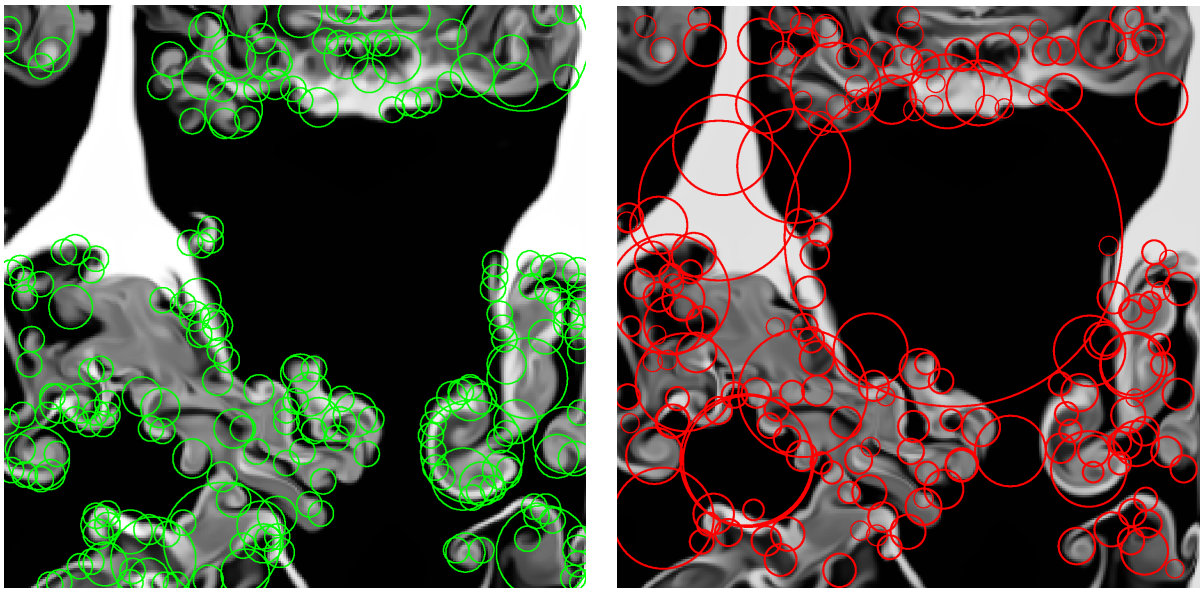
Figure 1 shows a comparison between the two methods that we focus on: Scale Saliency [4] and difference of gaussians [6]. Scale Saliency works by looking for regions where the distribution of grayscale values is somewhat unique to that scale. The difference of gaussians effectively involves convolving the image with a difference of two gaussians: one with a larger variance and one with a smaller variance. Then regions are selected as extrema in the scale and translational spaces.

Figure 2 shows the effect that additive white gaussian noise has on the Scale Saliency distinct region detector. In the noisier images the detector produced significantly more regions. To the first order, more detections are better: more possiblities to search over, less chance that a region is missed entirely.

## 2.3 Orientation Assignment

After the distinct region have been selected, they are intrinsically described by a local coordinate system with a well-defined scale and translation. However, we also desire rotation invariance. Instead of using a rotation invariant descriptor, following the work of Lowe [6] we will estimate the rotation (term used almost interchangeably with orientation). Specifically, to do this, we compute image gradient estimates for each pixel in the local region and then using linear weights place them into the two closest bins in a 36 bin histogram.

To select the orientation, we select the bin with the largest value and use a quadratic regression to find the orientation to sub bin accuracy. However, this method isn't robust. If a region has two dominant directions, then small variations will cause the orientation to vary widely. Instead, after picking the dominant orientation, we search for other orientations that have a bin value of at least 80% of the maximum and report these as separate orientations. For the rest of this paper, we will regard these alternate orientations as additional distinct regions. Our experiments suggest that lowering this threshold may be beneficial, since the orientation estimate



(a) (b)

Figure 1: An example of two distinct region detectors. (a) shows features found using Scale Saliency while (b) shows features found using Differences of Gaussians.

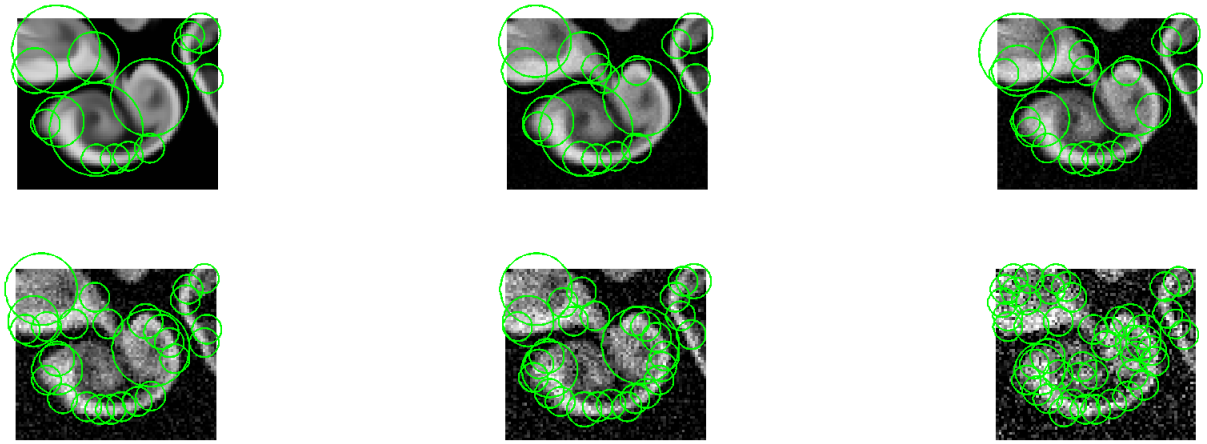
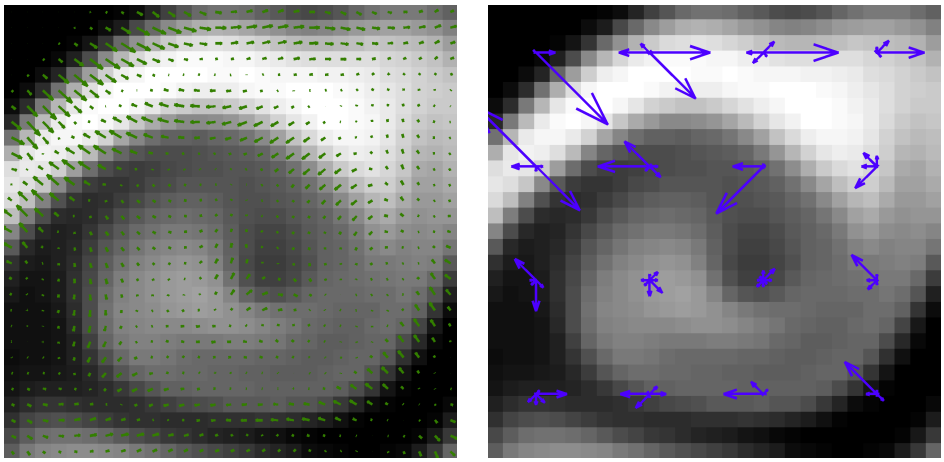


Figure 2: In order for our shape comparison to work well, we need a distinct region detector that is robust to common image changes. The upper left image is the original and the following five have progressively more additive white gaussian noise. As the amount of noise increases, so does the number of features detected. As long as the same regions are detected, the detection of additional regions is not an issue.



(a) (b)

Figure 3: (a) Per pixel image gradients computed using local differences. (b) The SIFT descriptor for this image. The SIFT descriptor uses a 4x4 grid of histograms, where directions are quantized to 8 bins. The overall descriptor contains 128 bins, and is robust to small translations and warpings of the image region.

is unreliable and additional distinct regions do not degrade performance significantly. Figure 3 (a) shows an example image with the pixel gradients overlayed.

### 2.4 Feature Descriptors

The feature descriptor that we use for this paper is the SIFT descriptor [6]. A recent comparison of local region descriptors showed that the SIFT descriptors are highly reliable [7]. The SIFT descriptors are a lower dimensional representation of an image region that is robust to common imaging artifacts.

The SIFT descriptors are computed using a histogram of image gradients. Figure 3 (a) shows the image gradients and (b) shows the histograms. Following the Lowe implementation, we divide the region into 16 regions organized as a 4x4 grid. For each region we split the directions into 8 separate groups. Each image pixel gradient is linearly divided by the closest regions and the closest directional bins. Once the histogram has been computed, it is normalized to unit length, thresholded to a maximum of 0.2 and then renormalized. We use the resulting 128 dimensional vector for the remainder of the paper.

## 3 Method

Our method for shape analysis relies on using a small number of descriptors to estimate a geometric transformation. The basic procedure for our method is outlined in psuedo-code in Figure 4.

The input is a box around a region of interest and the output is a list of regions that contain similar shapes, each with a corresponding shape score. Our shape score contains two terms: a descriptor cost term $c_d$ and a

```
find distinct regions
generate descriptors
pick descriptors in region of interest (source)
search for descriptor matches in target images
for each match do
    estimate transformation
    find matches for other points in region
    recompute transformation
    assign shape score
end
sort results by score
```

Figure 4: Psuedocode for searching for a shape.

geometry term $c_g$. The descriptor cost term depends only on the two sets of descriptors. Because the ordering of the SIFT descriptors seems to be a better indication of similarity than the relative distance, we adopt a descriptor cost based on $p_{ij}$, the index of descriptor $j$ in the sorted list of features similar to descriptor $i$. Using $K_d$ as an arbitrary constant and $\mathcal{C}$ as the set of correspondences $(i, j)$, our descriptor cost is:

$$c_d = \sum_{(i,j)\in\mathcal{C}} K_d(p_{ij})^2 \tag{1}$$

In turn, the transformation cost depends on three separate costs: translation, rotation and scale. Our transformation is a matrix $A$ that consists of a rotation($A_r$), translation and scale($A_s$) change between the source and target shape. The translation cost is the error in predicting locations. Using $x_i$ as the location, $r_i$ as the orientation and $s_i$ as the scale of descriptor $i$:

$$c_g^t = \sum_{(i,j)\in\mathcal{C}} \|Ax_j - x_i\|^2 \tag{2}$$

$$c_g^s = \sum_{(i,j)\in\mathcal{C}} K_s(\frac{s_i}{s_j} + A_s\frac{s_j}{s_i})^2 \tag{3}$$

$$c_g^r = \sum_{(i,j)\in\mathcal{C}} K_r((A_r + r_j - r_i) \bmod 2\pi) \tag{4}$$

Combining all of these costs, we get a shape cost:

$$c(\mathcal{C}) = c_d + c_g^t + c_g^r + c_g^s \tag{5}$$

### 3.1 Finding and describing features

Following the description in the previous section, we try out two separate tools to find distinct regions: Difference of Gaussians [6] and Scale Saliency [4]. Each of these detectors produces an (x,y) location and a radius - effectively describing a circular region.

After finding the features, we follow the procedure of Lowe: estimate an orientation and describe the local region using a histogram of image gradients. Details can be found in [6]. Empirically we have found that the orientation estimate is somewhat unreliable, and correspondingly set the threshold for multiple orientations lower.

### 3.2 Searching for features

The SIFT descriptors emit a 128 dimensional vector that describes the local region. Our basic search strategy is to search through the entire database of features for ones that are similar to ones in our region of interest. Specifically, we search for the K most similar regions for the N largest features in the input region.

Depending on experiment size, we typically chose K to be 50 and N to be 5. In other words, for the 5 largest features we picked the 50 closest matches to be candidate locations for a similar shape.

The goal of this stage is a wide net that will capture all similar shapes. It doesn't need to be very accurate, but it shouldn't miss many similar items. Empirical evidense suggests that on this problem size, these numbers are large enough.

### 3.3 Computing a rough transform

Since each of the local regions come with a local coordinate system, one can estimate translation, rotation and scale changes simply based on a single feature correspondence. While this estimate of local coordinate system is not particularly accurate, it is just a initial estimate.

### 3.4 Finding other matches

The shape comparison relies on having a group of features in correspondence between the query region and the result region. So far, we have computed the similarity simply based on a single correspondence. To compute a larger number of correspondences in the same region, we compute the correspondence cost on a per descriptor basis. Since the number of descriptors at small sizes increases dramatically, we limit the minimum size of the descriptors that can match. Finally, we pick the local M matches with the smallest cost and denote the correspondence as $\mathcal{C}$.

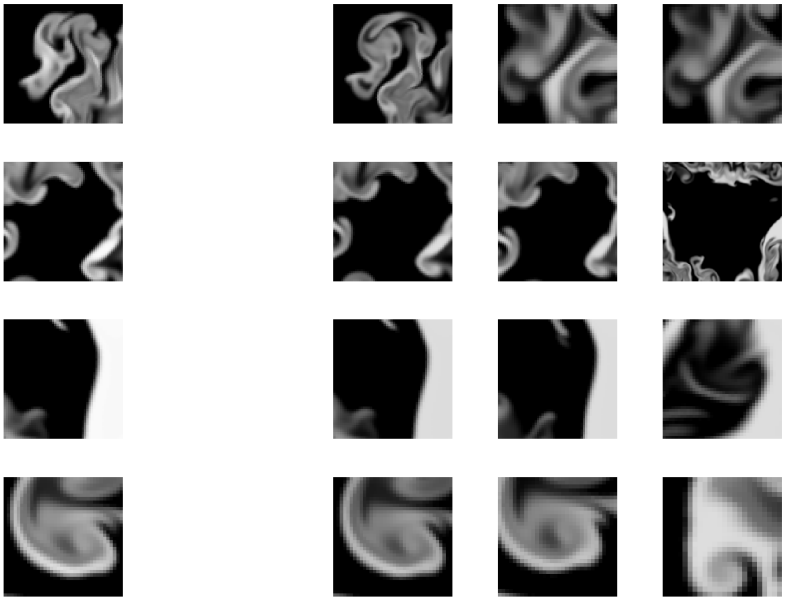


Figure 5: In order to match regions, the first step is to match features using the SIFT descriptors. Searching using SIFT descriptors tends to produce good results in common situations [7]. The column on the left are the query descriptors and the columns on the right are the top three matches.

### 3.5 Recomputing the transformation

Using the labeled correspondence $\mathcal{C}$ between descriptors in the query region and the target region, we recompute the transformation in order to get a more accurate shape score. To compute the transformation, we minimize the geometric cost term by running gradient descent, with the unknown variable effectively being the transformation A. However, since the space of legitimate transformations is smaller than the space of all matrices A, we run gradient descent on 4 parameters: rotation, scale and two translation terms. In most cases, gradient descent significantly improves our results over the initial estimate.

One could imagine iterating between the neighborhood descriptor searching and the computation of the transformation to produce an even better result. At this point in time, these tests have not been run.

### 3.6 Computing and sorting the score

Finally, once the correspondence between descriptors $\mathcal{C}$ has been selected and the transformation $A$ has been solved for, we can compute a score for each possible region matchusing equation 5. Once all of the scores have been computed, we can sort the results in several different forms. First, we can try to compare two regions, and give a comparison score as the best match of the first region in the second region. Alternatively, if we were searching for a similar shape, we could order the results as the most similar shapes.

# 4 Experiments

In order to gauge the effectiveness of this method of shape matching, we run a number of experiments, trying this technique in various settings to estimate how well it responds to different types of image alterations.

The goal of this technique is not to be invariant to geometric transformations or the addition of noise, but rather to be robust to these alterations. A warped image of fluids mixing is not the same as an unwarped image. Rather, the two images are similar. Instead, we look for graceful degredation: as the degree of the alteration increases, the metric should report a larger and larger disparity score.

However, a warped image should be still be similar to the original image - more similar than other unrelated images. To judge this, we search over a database of fluid images, showing that the best results come from similar images.

## 4.1 Geometry Based Experiments

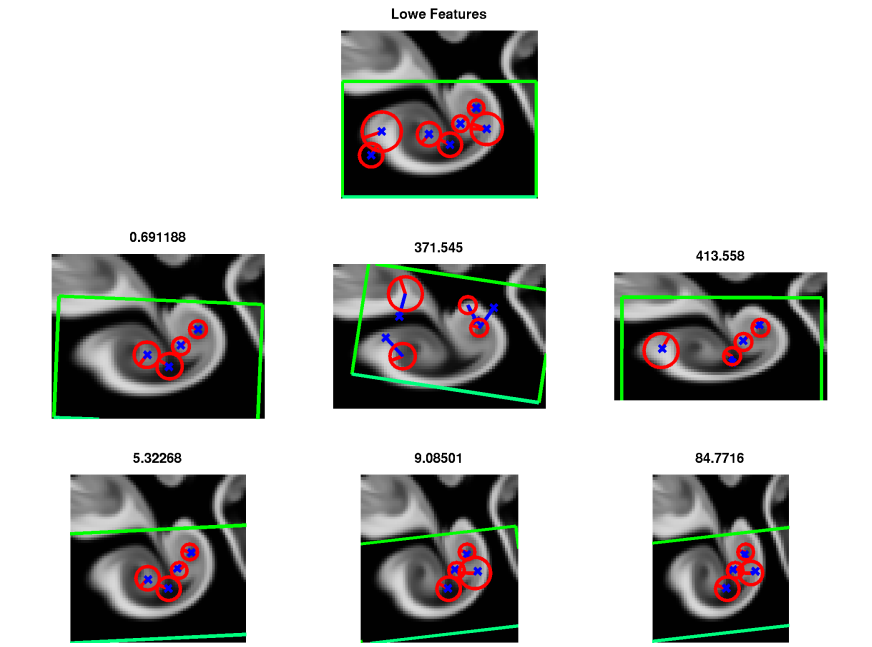
In the first batch of experiments, we look for graceful degredation of the shape score with respect to simple geometric transformations. Specifically, we artificially modify the images by shrinking them in the x and y directions separately. We use this as a simple comparison case and find that the Scale Saliency distinct region detector outperforms the Difference of Gaussians. In the Scale Saliency case, the results are as expected: images with smaller transformations produces smaller shape dissimilarity scores. Results are shown in Figure 6.

## 4.2 Search Based Experiments

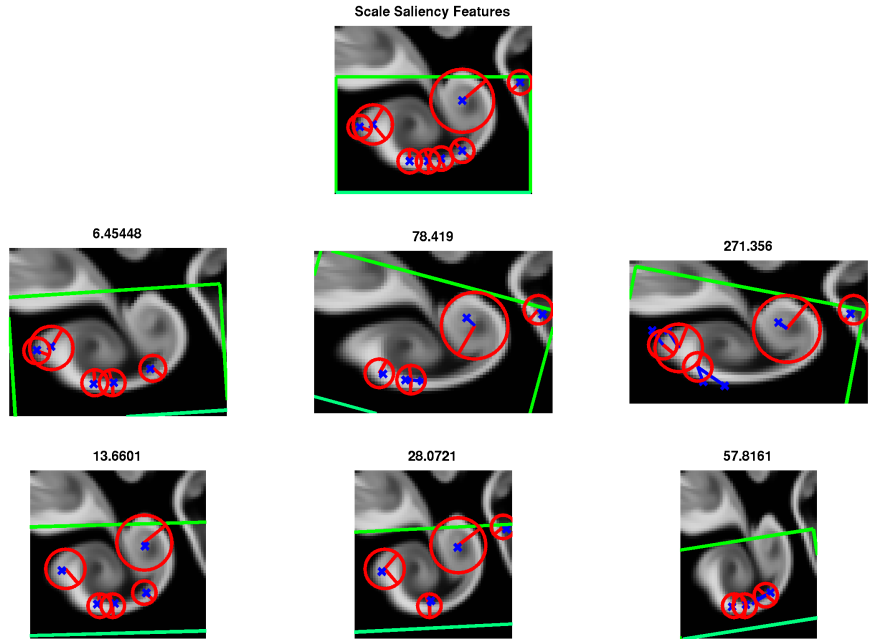
Another way to test the validity of our shape model is to search for similar shapes. In the database of 9 images that we have, we searched on a small region in one image to find other similar regions. As shown in Figure 7, using Difference of Gaussians as a region detector, the top two matches are the most similar regions in the dataset.

## 4.3 Noise Based Experiments

Ideally, a shape dissimilarity score not only responds well to geometric change, but also to addition of noise. We consider this to specifically be a strong point of our method in reference to other shape methods that use techniques like the medial axis transform. Again, results are pleasing and illustrated in Figures 8 and 9. Note that the parameter settings on the shape dissimilarity score are different from the experiments in the geometry section and don't align perfectly.




(a)




(b)

Figure 6: In order to test our model of shape, we 'query' using the top image and then look for results in manually transformed images. The first row of images are shrunk in the vertical direction, while the second row is shrunk in the horizontal direction (by 10%, 20% and 30% respectively). Ideally, the images with smaller transformations will have lower scores (shown above the image). (a) was generated using Difference of Gaussians while (b) uses Scale Saliency.

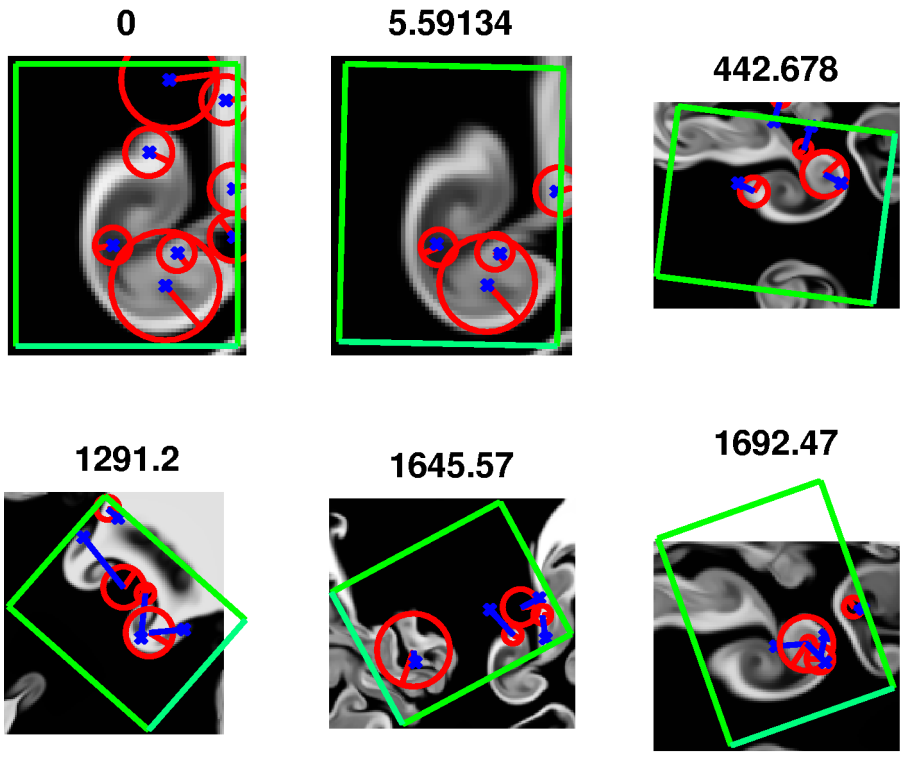



Figure 7: A search based experiment using difference of gaussians. The first image is the query region and the next 5 are the results. There were only 3 instances of this shape in the dataset, so the last 3 results are shapes that our algorithm ranked as similar.

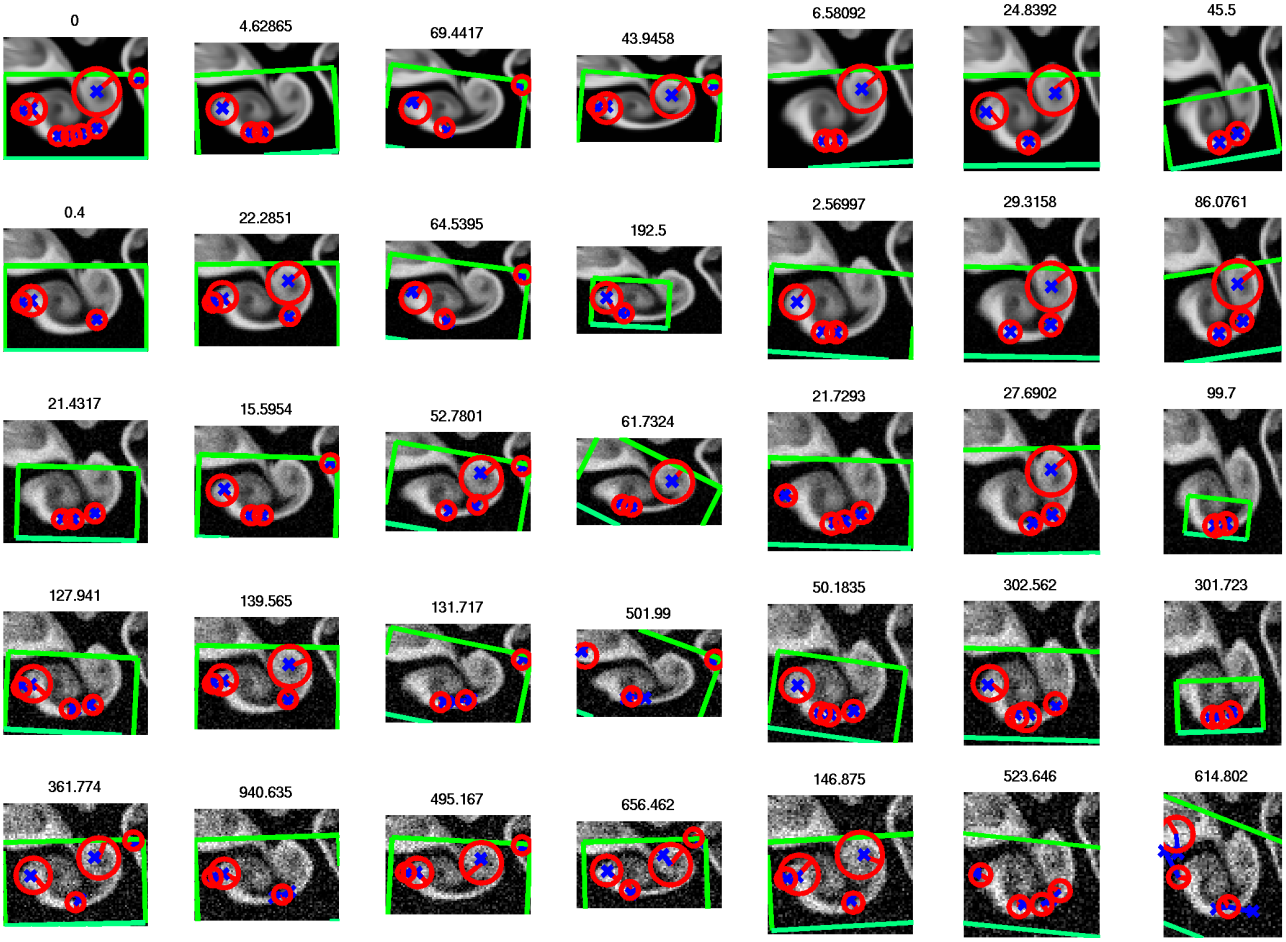



Figure 8: A comparison of scores under our shape comparison for images that have been adjusted geometrically and had noise added to them. The following figure shows the same results re-ordered by shape score. Scale saliency was used to generate the distinct regions.

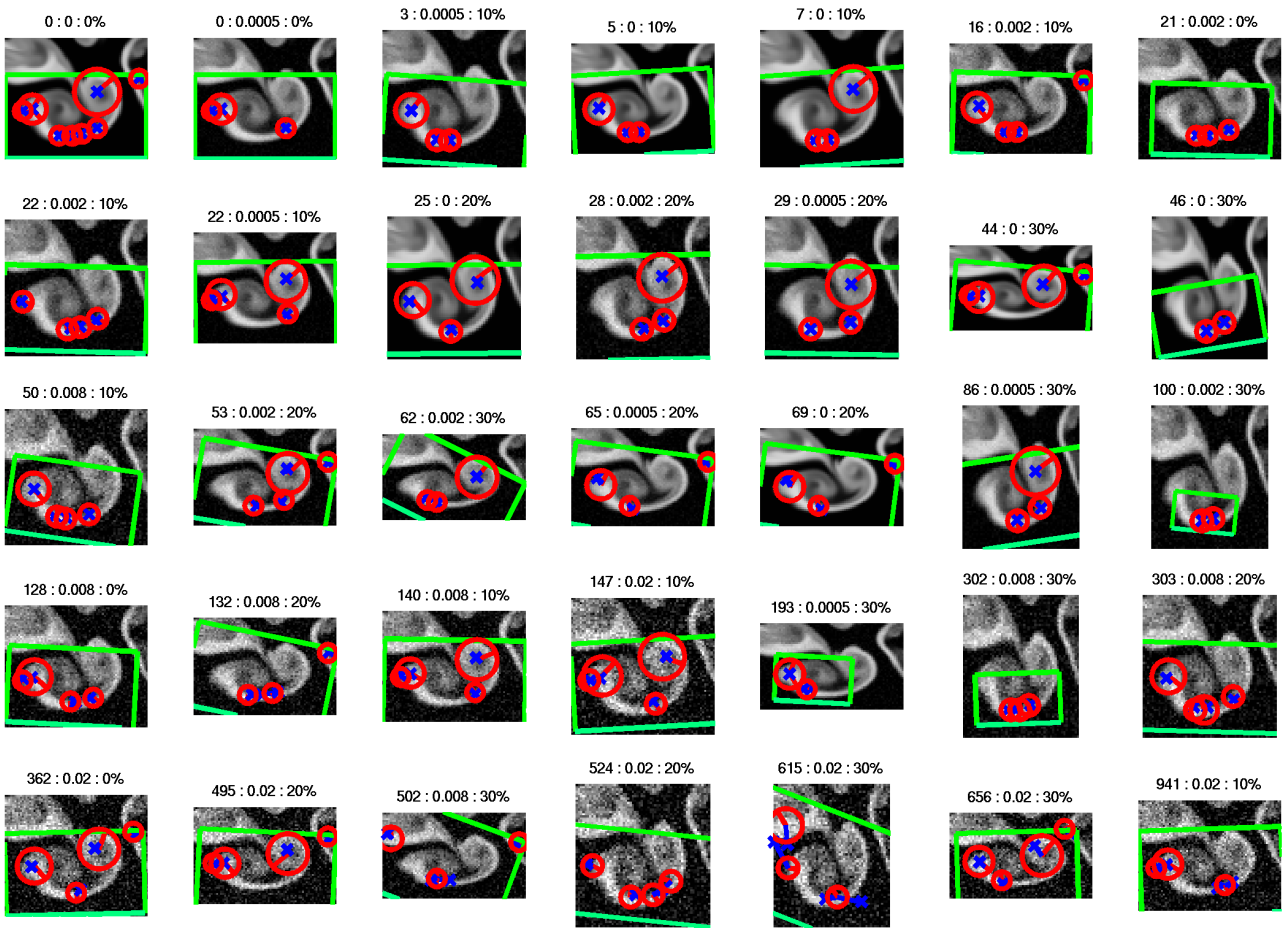



Figure 9: Similar to the previous slide, this shows images corrupted by noise and geometric transformations sorted by shape score. The upper left image is the original, the region detector is Scale Saliency and titles are (score : noise : geometric warp)

# 5 Discussion

Our shape score algorithm is successful on this dataset - it is robust to geometric deformations and noise. Furthermore, it is invariant to translation, rotation and scaling. While untested, we claim that it is also robust to occlusion - since there is no requirement about continuity. This is in strong contrast to previous work on shape that requires contiguous shapes, presented as black and white images.

However, despite the strong aspects of our work, we should note some shortcomings. First, our algorithm is somewhat sensitive to choices in parameters - how many regions to search for and how many to match. Ideally, we would like to develop an automatic way to set these numbers. Correspondingly, it would be useful to emphasize the area that the distinct regions cover in the shape. More area would indicate a better match.

The weakest part of this work is probably the use of SIFT descriptors. These descriptors are highly descriminative, but don't deal well with similar, though not identical regions. We tried to remedy this by using the ranking of the descriptors rather than the distances, but feel a stronger method would work better. Shape context [1] is a good possibility here. Alternatively, one could imagine using thresholds to limit the SIFT distances.

Finally, since the feature correspondence is solved in a simple manner, the correspondence is not one-to-one. Some features are doubled matched - an obvious error in logic. More advanced techniques could fix this.

**Acknowledgments**:

Ryan White was supported by a Department of Homeland Security Graduate Fellowship. Image data was provided by the ASCI Turbulence Project.

UCRL-TR-200271. This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

# References

[1] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape context: A new descriptor for shape matching and object recognition. In *NIPS*, pages 831–837, 2000.

[2] David Forsyth and Jean Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 2003.

[3] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[4] Timor Kadir and Michael Brady. Saliency, scale and image description. *International Journal of Computer Vision*, 2001.

[5] Alireza Khotanzad and Yaw Hua Hong. Invariant image recognition by zernike moments. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(5), 1990.

[6] David Lowe. Distinct image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2004.

[7] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[8] Greg Mori, Serge Belongie, and Jitendra Malik. Shape contexts enable efficient retrieval of similar shapes. In *Computer Vision and Pattern Recognition*, 2001.

[9] Fred Rothganger, Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. 3d object modeling and recognition using affine-invariant patches and multi-view constraints. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[10] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or "How do I organize my holiday snaps?". In *Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark*, volume 1, pages 414–431. Springer-Verlag, 2002.

[11] Thomas Sebastian, Philip Klein, and Benjamin Kimia. Recognition of shapes by editing shock graphs. In *Proceedings of the International Conference of Computer Vision*, pages 755 – 762, 2001.

[12] Remco Veltkamp and Michiel Hagedoorn. State-of-the-art in shape matching. Technical report, Utrecht University, 1999.

[13] Dengshend Zhang and Goujun Lu. A comparitive study of three region shape descriptors. In *Digital Image Computing Techniques and Applications*, 2002.